Technical Report

Nr. TUD-CS-2011-0104
April 6th, 2011

CASED

CertainLogic: A Logic for Modeling Trust and Uncertainty

Authors

Sebastian Ries[1], Sheikh Mahbub Habib[1], Max Mühlhäuser[1] and Vijay Varadharajan[2]

[1] Technische Universität Darmstadt, Darmstadt, Germany

[2] Macquarie University, Sydney, Australia

CertainLogic: A Logic for Modeling Trust and Uncertainty

Sebastian Ries[1], Sheikh Mahbub Habib[1], Max Mühlhäuser[1], and Vijay Varadharajan[2]

[1] Technische Universität Darmstadt, CASED
Darmstadt, Germany
[2] Macquarie University
Sydney, Australia

Abstract. The evaluation of the trustworthiness of complex systems is one of the major challenges in current IT research. We contribute to this field by providing a novel model for the evaluation of propositional logic terms under uncertainty that is compliant with the standard probabilistic approach and subjective logic. Furthermore, we present a use case to demonstrate how this approach can be applied to the evaluation of the trustworthiness of a system based on the knowledge about its components and subsystems.

1 Introduction

The evaluation of the trustworthiness of complex systems is one of the major challenges in current IT research, as – following the visions of the Internet of Services, the Future Internet and Cloud Computing – IT systems become highly distributed, dynamically composed, and hosted and managed by multiple parties. For example, in the field of Cloud Computing (which we will take as a running example in the following), people and enterprises are still hesitating to *move to the Cloud* due to missing transparency and security concerns [1–3]. However, it is not only the users who are interested in evaluating the trustworthiness of a service, infrastructure, or platform, but also the providers and accreditation authorities.

Currently, there are several approaches supporting those stakeholders in assessing the trustworthiness of such kind of systems, e.g., from the field of trusted computing [4], experience-based trust and reputation models [5], and security [6–8]. The first two approaches tackle the evaluation of trust from opposite sides. Trusted computing may provide a technical root of trust, however, attestation concepts based on trusted computing, e.g., [9, 10] focus on the evaluation of single platforms not on compositions. On the other hand experience-based trust models concentrate on deriving the trustworthiness of a service or system from user feedback without considering technical details. In contrast, the approaches in [6–8] require knowledge on the implementation or address a particular security requirement only.

However, for complex systems there is a lack of models that provide means for deriving the trustworthiness of the overall system considering (1) the trustworthiness of the subsystems and atomic components (independently from how these trust values are assessed), (2) information on how the system combines its subsystems and components, and (3) the knowledge about which subsystems and components are redundant.

Hereby, a major challenge of this task is taking into account that in real world applications this information about the trustworthiness of the subsystems and components itself is subject to uncertainty. For example, reputation values might be based on insufficient information and current solutions from the field of trusted computing cannot effectively capture dynamic changes in trust [11]. Also when considering the recent advances in the field of property-based attestation (e.g., [9]), there is a need for modeling trust and uncertainty in order to deal with the fact that (1) the state of the system that was measured at the time of booting does not necessarily reflect the state of the system at the time of attestation and (2) that the authority that provides the property certificates might only be trusted to a certain degree [12]. Thus, models for evaluating the trustworthiness of a complex system need to be capable of modeling the uncertainty that is associated to the trustworthiness of the subsystems and components of the system and to also calculate and express the degree of uncertainty associated to the derived trustworthiness of the overall system.

In this paper, we provide a model for the evaluation of propositional logic terms under uncertainty. The model has been designed to be compliant to the standard probabilistic approach and subjective logic [13,14], which also provides the justification for the mathematical validity of the model. In our model a statement about the truth of a proposition explicitly takes into account the initial expectation about the truth of this proposition and the knowledge acquired in previous experiments - including a parameter that reflects uncertainty associated to this knowledge. In contrast to subjective logic, our model is based on independent parameters, which is considered to be an advantage from the modeling perspective and which allows for a more intuitive representation. As a core contribution of this paper, we describe how the parameters of the model can be assessed; we define operators for AND, OR, and NOT such that the uncertainty can be reflected in the input as well as in the calculated result; and we give the properties of these operators. Furthermore, we introduce a use case to show how this approach could be used for evaluating the trustworthiness of a system in a Cloud Computing scenario and to show how the evaluation of the trustworthiness of a complex system relates to the evaluation of propositional logic terms. However, neither the representation of propositions nor the evaluation of the operators are restricted to this use case.

The remainder of the paper is structured as follows: First, we clarify the terminology (Sec. 2), present the related work (Sec. 3), and introduce a use case (Sec. 4). Next, we present the model (Sec. 5), give some examples (Sec. 6), and show the compliance with subjective logic (Sec. 7). Finally, we discuss the evaluation of the use case (Sec. 8), and draw our conclusions (Sec. 9).

2 Terminology

In the following, we briefly introduce our understanding of trust and uncertainty.

Trust & Probabilities Trust is a well-known concept in everyday life and often serves as a basis for decisions in complex situations, and there have been numerous approaches for modeling this concept in different research fields of computer science, e.g., virtual organizations [15–17], mobile and P2P networks [18,19], and eCommerce [20,21]. Although, the general concept of trust is well-known, it is hard to give a definition. A definition, which is currently shared or at least adopted by many researchers [5,15,16,22], has been provided by the sociologist Diego Gambetta [23]: "trust [...] is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action [...] and in a context in which it affects its own action." Following this definition, we also consider trust as a subjective probability, which can be modeled as Bayesian probabilities [24], and which is - when derived from previous experience - subject to uncertainty. However, we extend the scope from agents to software or computer systems in general.

Uncertainty For making our understanding of uncertainty more explicit, we refer to a simple example with *dice*. For a *Laplace* die, it is not possible to say whether it will show a "6" when thrown the next time, but it is known that the probability for showing a "6" is 1/6. Although, in this case the outcome of the next throw is uncertain, there is no uncertainty associated to the probability. In contrast, for a *real* die the latter must not be true. When given a *real* die, one could assume that the probability for showing a "6" is 1/6 based on a subject's prior knowledge about dice, however, this statement is still subject to uncertainty, as the die could have been manipulated. In order to reduce the uncertainty, one could throw the die a number of times, 5 times, 10 times, 100 times, ... based on the assumption that this leads to more and more representative estimates about the probability for showing a "6". Formally, this could be modeled using Bayesian probabilities (see [24] Chap 1 & 4 for a discussion of the Frequentists interpretation and the Bayesian interpretation of probabilities). On the other hand, instead of throwing the die one could also examine it and say based on one's expert knowledge, one is quite certain that the probability for a "6" is 1/6, or a non-expert could say "I guess the probability is 1/6, but I am not really certain about this guess". In this paper we are focusing on the latter type of uncertainty, where uncertainty is associated to the probabilities under evaluation and relates to the question whether the past experience is representative for the future behavior.

3 Related Work

In the field of trust modeling there is a number of approaches modeling the (un-)certainty of a trust value, well-known approaches are given in [16, 17, 19–21, 26, 27]. However, those approaches do not tackle the issue of deriving the trustworthiness of a system based on the knowledge about its subsystems and components, instead the challenge of these approaches is to find good models for deriving trust from direct experience of a user, recommendations from third parties, and also additional information, e.g. social relationships. Especially, those models aim on providing robustness to attacks, e.g., misleading recommendations, re-entry, Sybil attacks, etc. (see also [5, 28, 29] for a survey of attacks). For those tasks they usually provide operators for combining evidence from different sources about the same target (also called consensus or aggregation) and for weighting recommendations based on the trustworthiness of the source (also called discounting or concatenation). However, the goal of those approaches is not to provide operators for the evaluation of propositional logic terms.

Although, there are researchers in the field of trust focusing on modeling (un-)certainty [13,16,22,30], they do not provide operators for the evaluation of propositional logic terms, except for "subjective logic" [13, 14].

Finally, there are well-known approaches for modeling uncertainty outside the trust field. At first, there is the standard probabilistic approach. However, this approach only allows to deal with the uncertainty of the outcome of the next event, but probabilities are assumed to be known.

Furthermore, fuzzy logic [31] seems to be related, however, it models another type of uncertainty, which could be typed as linguistical uncertainty or fuzzyness. For example, if it is "hot" in a room with a degree of 0.8, it does not mean that the probability that it is hot in this room is 80% (assuming that being hot means temp $> 30°$); but it means that one cannot agree on a clear threshold when it is hot (above we assumed $30°$), and thus a degree of 80% states that it is closer to hot than to cold.

There is the field of (Dempster-Shafer) belief theory, which again leads to "subjective logic" [13]. The main drawback of this model is that the parameters for *belief*, *disbelief*, and *uncertainty* are dependent on each other, which introduces an unnecessary redundancy from the perspective of modeling and prevents one from re-assign just a single parameter.

Beyond subjective logic there are numerous other approaches for probabilistic reasoning, for further references see e.g. [32]. However, as we argue for the mathematical validity of our model based on its compliance to subjective logic and the standard probabilistic approach, we do not provide a discussion of probabilistic reasoning in general.

Finally, it is possible to model uncertainty using Bayesian probabilities [24], this usually leads to probability density functions, e.g., the Beta probability density function. For the approaches in [13, 22], it has been shown that there are bi-directional mappings between the representations proposed in those papers and the Beta probability density function. It is possible to apply the propositional standard operators to probability density functions, however, this leads to complex mathematical operations and multi-dimensional distributions, which are also hard to interpret and to visualize. In our proposed approach, we will not increase the dimensions when calculating AND and OR.

While our previous work [22,33,34] shares the basic representation with CertainLogic, our previous research focuses on the modeling of experience-based trust, e.g., providing means for *aging*, operators for *consensus* and *discounting*, and for coping with Sybil attacks. The approach of deriving the trustworthiness of a system based on its components and subsystems considering uncertainty, the discussion how to assess the parameters, as well as the definition of the operators for the evaluation of propositional logic terms and the introduction of their properties are original contributions of this paper.

4 Use Case

In the following, we introduce a scenario from the field of Cloud Computing, and show how the evaluation of the trustworthiness of the overall system can be carried out, if there is an appropriate approach for the evaluation of propositional logic terms (see also [25]).

Assume that we evaluate the trustworthiness of a simple Customer Relationship Management (CRM) system focusing on the availability of the system.

In the example (see Fig. 1), the CRM system S directly relies on two subsystems, S_1 providing authentication capabilities, S_2 offering storage capacity for sales data and data mining capabilities, and an atomic component C for the billing. Subsystem $S1$ consist of two authentication servers (A_1 and A_2), where at least one of the servers has to be available. Similarly, subsystem $S2$ is composed of three redundant data bases servers (only one needs to be available).

Based on the description above and assuming that the information about the trust values of the atomic components is known, the evaluation of the trustworthiness of the complete system in the context of availability, can be carried out by evaluating the following propositional logic term:

$$(A_1 \vee A_2) \wedge (B_1 \vee B_2 \vee B_3) \wedge C$$

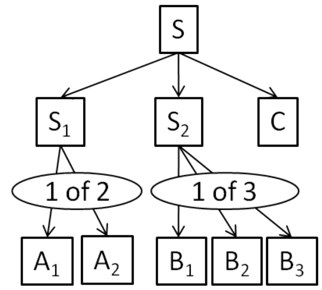


Fig. 1. System architecture (incl. information about redundant components)

where A_1 is a proposition that is true if the component A_1 behaves as expected (e.g., the component replies to requests within a certain time limit); the interpretations of the other propositions are assigned in the same way. Although, we restricted the scope of our example to availability, please note that it is possible to model statements about the fulfillment of other relevant properties (e.g., attested / self-evaluated security properties or reputation of a component or subsystem) as propositions and to consider them in the evaluation of the overall trustworthiness of the system using propositional logic terms. However, as the knowledge about the fulfilment of the propositions is subject to uncertainty, the evaluation method has to take this uncertainty into account when calculating the trustworthiness of the overall system.

5 CertainLogic

In the following, we introduce a novel model, which we call *CertainLogic*, for evaluating propositional logic terms that are subject to uncertainty. Especially, we define the standard operators of propositional logic: AND, OR, and NOT. However, before introducing these operators, we have to introduce a way for modeling probabilities and uncertainty.

5.1 CertainTrust - Representation

The model for expressing opinions, this is how we call the construction for modeling probabilities that are subject to uncertainty (in accordance with [13]), is called *CertainTrust* [22]. CertainTrust (CT) bas been designed as a representation for evidence-based trust, but may also serve as a representation for uncertain probabilities. Additionally, it supports users with a graphical, intuitively interpretable interface (see Table 2).

Definition 5.1 (Representation CertainTrust)
In CertainTrust, an opinion o_A about the truth of a proposition A is given as $o_A = (t, c, f)$ where the parameters are called average rating $t \in [0, 1]$, certainty $c \in [0, 1]$, *and* initial expectation value $f \in]0, 1[$[3]. *If it holds $c = 0$ (complete uncertainty), the expectation value (see Def. 5.2) depends only on f, however, for soundness we define $t = 0.5$ in this case.*

[3] We exclude $f = 0$ and $f = 1$ here for the sake of simplicity when defining the operators.

The following introduces the basic semantics of the parameters[4]. The *average rating* t indicates the degree to which past observations (if there are any) support the truth of the proposition. It can be associated to the relative frequency of observations supporting the truth of the proposition. The extreme values can be interpreted as follows:

- average rating = 0: There is only evidence contradicting the proposition.
- average rating = 1: There is only evidence supporting the proposition.

The *certainty* c indicates the degree to which the average rating is assumed to be representative for the future. It can be associated to the number of past observations (or collected evidence units). The higher the certainty of an opinion is, the higher is the influence of the average rating on the expectation value in relation to the initial expectation. When the maximum level of certainty ($c = 1$) is reached, the average rating is assumed to be representative for the future outcomes. The extreme values can be interpreted as follows:

- certainty = 0: There is no evidence available.
- certainty = 1: The collected evidence is considered to be representative.

The *initial expectation* f expresses the assumption about the truth of a proposition in absence of evidence.

In Section 5.2, we describe different ways to assess the parameters (i.e. t, c, and f).

Definition 5.2 (Expectation value of CT)
The expectation value *of an opinion* $E(t, c, f) \in [0, 1]$ *is defined as* $E(t, c, f) = t * c + (1 - c) * f$.

It expresses the expectation about the truth of the proposition taking into account the initial expectation, the average rating and the certainty.

5.2 Assessment of the parameters

The parameters for an opinion $o = (t, c, f)$ can be assessed in multiple ways.

Direct assessment: They can be assessed directly, e.g., based on the opinion of an expert, who estimates initial expectation value f based on her overall knowledge of the topic, the average rating t is derived from the available data, and the certainty c expresses the expert's confidence in the representativeness of the average rating.

Derived from direct experience and recommendations: They can be derived from a trust or reputation system that takes into account one's past experience and recommendations from third parties. Especially, CertainLogic can directly be applied to trust or reputation values of Bayesian trust models, as e.g. [16, 19, 21, 26]. Furthermore, when considering that those models provide operators for discounting, those models can be used to increase the uncertainty when the information about the truth of a statement is received from a source that is not fully trusted.

[4]There are additional parameters defined in [22], i.e., the *weight* w of the initial belief, the *number of expected evidence units* N, and a parameter for considering the *age* of evidence. When deriving the parameters (t, c, f) from past evidence, one could assume $w = \frac{1}{2}(r_0 + s_0) = 1$ and $\lim N \to \infty$. The parameter *age* is not directly relevant for this paper.

Subjective Logic: They can be derived from an opinion given in subjective logic (see Section 7).

Beta probability distribution: The Beta distribution is a commonly used distribution for a random variable $0 \leq p \leq 1$. The Beta probability density function $f(p \mid \alpha, \beta)$ can be given as:

$$f(p \mid \alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}, \quad \text{where } 0 \leq p \leq 1,\ \alpha > 0,\ \beta > 0\ . \tag{1}$$

By defining $\alpha = r + r_0$ and $\beta = s + r_0$, it is possible to relate the probability function directly to the collected evidence (or observed outcomes), where r and s represent the number of outcomes supporting the proposition and contradicting the proposition, respectively, and r_0 and s_0 define the prior knowledge ($r_0 + s_0 \neq 0$) (see [22, 35]). The expectation value is defined as $E = \frac{\alpha}{\alpha+\beta}$. The mathematical foundations of this Bayesian approach are described, e.g., in [24].

Definition 5.3 (Mapping CT to Beta pdf)

Using the Beta probability density functions's (Beta pdf's) input parameters, an opinion can be denoted as (r, r_0, s, s_0). It can be mapped[5] to CertainTrust using $(t, c, f) = m^{B}_{CT}(r, r_0, s, s_0)$, where it holds:

$$\begin{aligned} t &= \begin{cases} 0.5 & \text{if } r+s=0\ , \\ \frac{r}{r+s} & \text{else}\ . \end{cases} \\ c &= \frac{r+s}{r+s+2} \\ f &= \frac{r_0}{r_0+s_0} \end{aligned} \tag{2}$$

For $c \neq 0$ the mapping can also be calculated in the inverse direction (as a simplified version of the description given in [36] pp. 75 & 86), however for brevity, we do not present the inverse mapping here.

The mapping between CertainTrust and Beta pdfs is especially interesting, as both representations have been shown to calculate the same expectation value [22, 36], i.e., it holds $E(t, c, f) = E(f(p|r + r_0, s + s_0))$ if $(t, c, f) = m^{B}_{CT}(r, r_0, s, s_0)$.

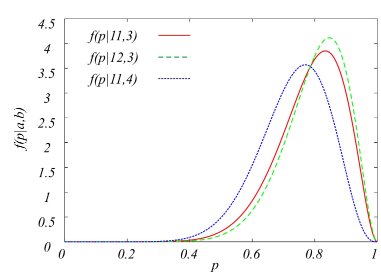


Fig. 2. Beta probability density function

5.3 Logical Operators

Having introduced the representational model, and explained how the parameters can be assessed, we define the operators of propositional logic (OR, AND, and NOT). These operators are defined in a way that they are compliant with the evaluation of propositional logic terms in the standard probabilistic approach. However, when combining opinions, those operators will especially take care of the (un-)certainty that is assigned to its input parameters, and reflect this (un-)certainty in the result.

[5] In the following we use the convention m^{from}_{to} whenever defining a mapping.

Table 1. Definition of the operators

OR	$c_{A\vee B} = c_A + c_B - c_A c_B - \frac{c_A(1-c_B)f_B(1-t_A) + (1-c_A)c_B f_A(1-t_B)}{f_A + f_B - f_A f_B}$ $t_{A\vee B} = \begin{cases} \frac{1}{c_{A\vee B}}\left(c_A t_A + c_B t_B - c_A c_B t_A t_B\right) & \text{if } c_{A\vee B} \neq 0\ , \\ 0.5 & \text{else}\ . \end{cases}$ $f_{A\vee B} = f_A + f_B - f_A f_B$
AND	$c_{A\wedge B} = c_A + c_B - c_A c_B - \frac{(1-c_A)\,c_B\,(1-f_A)\,t_B + c_A\,(1-c_B)\,(1-f_B)\,t_A}{1 - f_A f_B}$ $t_{A\wedge B} = \begin{cases} \frac{1}{c_{A\wedge B}}\left(c_A c_B t_A t_B + \frac{c_A(1-c_B)(1-f_A)f_B t_A + (1-c_A)c_B f_A(1-f_B)t_B}{1-f_A f_B}\right) & \text{if } c_{A\wedge B} \neq 0, \\ 0.5 & \text{else}\ . \end{cases}$ $f_{A\wedge B} = f_A f_B$
NOT	$t_{\neg A} = 1 - t_A$, $c_{\neg A} = c_A$, and $f_{\neg A} = 1 - f_A$

Operator *OR* The operator OR is applicable when opinions for two independent propositions need to form a new opinion reflecting the degree of truth for at least one out of both propositions.

Definition 5.4 (Operator OR)
Let A and B be two independent propositions and the opinions about the truth of these propositions be given as $o_A = (t_A, c_A, f_A)$ and $o_B = (t_B, c_B, f_B)$, respectively. Then, the resulting opinion is denoted as $o_{A\vee B} = (t_{A\vee B}, c_{A\vee B}, f_{A\vee B})$ where $t_{A\vee B}$, $c_{A\vee B}$, and $f_{A\vee B}$ are defined in Table 1 (OR). We use the symbol $'\vee'$ to designate the operator OR and we define $o_{A\vee B} \equiv o_A \vee o_B$.

Operator *AND* The operator AND is applicable when opinions for two independent propositions need to be aggregated to produce a new opinion reflecting the degree of truth of both propositions simultaneously.

Definition 5.5 (Operator AND) *Let A and B be two independent propositions and the opinions about the truth of these propositions be given as $o_A = (t_A, c_A, f_A)$ and $o_B = (t_B, c_B, f_B)$, respectively. Then, the resulting opinion is denoted as $o_{A\wedge B} = (t_{A\wedge B}, c_{A\wedge B}, f_{A\wedge B})$ where $t_{A\wedge B}$, $c_{A\wedge B}$, and $f_{A\wedge B}$ are defined in Table 1 (AND). We use the symbol $'\wedge'$ to designate the operator $'AND'$ and we define $o_{A\wedge B} \equiv o_A \wedge o_B$.*

Operator *NOT* The operator NOT is applicable when an opinion about an proposition needs to be negated.

Definition 5.6 (Operator NOT)
Let A be a proposition and the opinion about the truth of this proposition be given as $o_A = (t_A, c_A, f_A)$. Then, the resulting opinion is denoted as $\neg o_A = (t_{\neg A}, c_{\neg A}, f_{\neg A})$ where $t_{\neg A}$, $c_{\neg A}$, and $f_{\neg A}$ are given in Table 1 (NOT). We use the symbol $'\neg'$ to designate the operator NOT and we define, $o_{\neg A} \equiv \neg o_A$

The operators for AND and OR can be shown to be commutative and associative. Both properties are essential for the evaluation of propositional logic terms.

Theorem 5.1 (Commutativity)
It holds $o_{A \wedge B} = o_{B \wedge A}$ *and* $o_{A \vee B} = o_{B \vee A}$

Theorem 5.2 (Associativity)
It holds $o_{A \wedge (B \wedge C)} = o_{(A \wedge B) \wedge C}$ *and* $o_{A \vee (B \vee C)} = o_{(A \vee B) \vee C}$.

The proofs are given in Appendices (A, B, and C).

The operators are *not distributive*, i.e., it does not hold that $o_{A \wedge (B \vee C)} = o_{(A \wedge B)} \vee o_{(A \wedge C)}$, as $A \wedge B$ and $A \wedge C$ are not independent propositions[6].

Compliance: Standard probabilistic approach In the standard probabilistic approach (which is not considering the uncertainty that might be associated with a probability), the operation for AND is usually defined as $p(A \wedge B) = p(A)p(B)$, the operation for OR is given as $p(A \vee B) = p(A) + p(B) - p(A)p(B)$, and the operation for NOT is given as $p(\neg A) = 1 - p(A)$.

The expectation value $E(t, c, f)$ of an opinion can be interpreted as the probability for the truth of a proposition and it can be shown that the following statements are true:

Theorem 5.3 (Compliance)
The propositional logic operators for AND, OR, and NOT as defined in Table 1 are compliant with the standard probabilistic evaluation of propositional terms as it holds:

1. $E(o_{A \wedge B}) = E(o_A)E(o_B)$ *(for AND)*
2. $E(o_{A \vee B}) = E(o_A) + E(o_B) - E(o_A)E(o_B)$ *(for OR)*
3. $E(o_{\neg A}) = 1 - E(o_A)$ *(for NOT)*

The proof is given in Appendix D.

Although, the standard probabilistic approach is compliant with CertainLogic, there are multiple advantages when combining opinions with CertainLogic:

– Our model can express the uncertainty, which is not possible in the standard probabilistic approach. This is important, because in real world scenarios probabilities are usually not known, but have to be estimated or derived from experiments, and thus they are subject to uncertainty.
– Our model does not only take the (un-)certainty as an input parameter, but it reflects also the uncertainty calculated for the result. Thus, the certainty is a good indicator for the confidence associated to the calculated result.

[6] The evaluation of propositional operators in the standard probabilistic approach does not satisfy *distributivity*, too, for the same reason.

6 Examples

In the following, we present some examples showing the impact of the newly defined operators on opinions modeled in CertainLogic. The left part of Table 2 shows 3 examples for the AND operator and the right part for the OR operator. For each example, we provide the opinions as tuple and additionally we provide the expectation value and the graphical representation of an opinion (and in the first row we also provide the parameters of the corresponding beta probability density functions[7]).

In the graphical representation the color-gradient indicates the expectation value of each point in the figure. Therefore, the color of each point in the figure is calculated as a linear combination of the RBG-vectors of red ($E = 0$), yellow ($E = 0.5$), and green ($E = 1$)[8]. For example, in the first row one can see how the AND operator affects the initial expectation f. Whereas for the initial expectation of A and B it holds $f_A = f_B = 0.5$, it holds $f_{A \wedge B} = 0.25$, as A and B have to be true simultaneously. This is directly reflected by the color-gradient. As the certainty of $c_A = c_B = 0$ (no experience available), the certainty of the resulting opinion is also $c_{A \wedge B} = 0$, and the expectation value of each opinion is equivalent to the initial expectation.

In the second row, we provide an example where we are certain ($c_A = 1$) that proposition A is false ($t_A = 0$) and for B it holds $c_B = 0$ (complete uncertainty). The certainty of the resulting opinion is $c_{A \wedge B} = 1$ and the rating is $t = 0$, as in this case the knowledge about A is sufficient to be sure that $A \wedge B$ is false.

However, the third row shows that if we are certain ($c_A = 1$) that proposition A is true ($t_A = 1$) and for B it holds $c_B = 0$ (complete uncertainty), then, the certainty of the resulting opinion is only $c_{A \wedge B} = 0.33$ as knowing that one proposition is true is not sufficient for AND.

The examples for the OR operator follow a similar reasoning. In the first row, one can immediately see how the initial expectation value is influenced by the OR operator; the resulting opinion's color-gradient is more 'greenish'. This is reasonable as the initial expectation value of the A and B are $f_A = f_B = 0.5$ and the resulting opinions expectation value is $f_{A \vee B} = 0.75$, as the chances that $A \vee B$ is true are higher than the chances for just A or just B.

7 Compliance to Subjective Logic

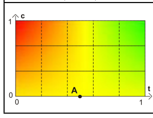
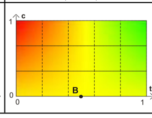
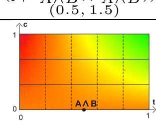
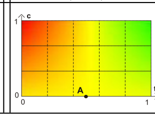
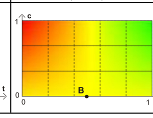
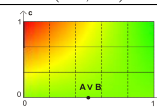
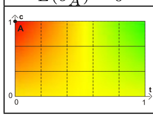
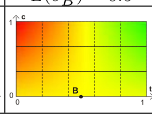
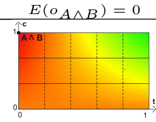
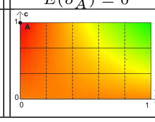
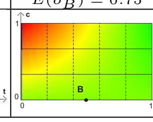
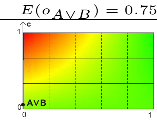
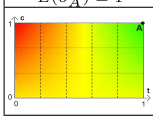
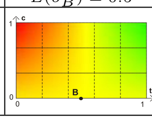
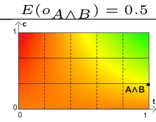
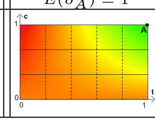
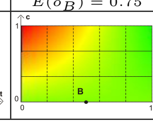
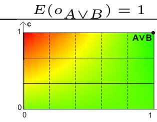
Subjective logic (SL) has been described in [13], and it combines elements from belief theory with Bayesian probabilities. In the following, we show that the operators of CertainLogic are compliant with those of subjective logic, which finally provides the argument for the mathematical validity of our approach.

Definition 7.1 (Belief representation (SL))
According to [13], an opinion is given by $\omega = (b, d, u, a)$*, where* b *models the* belief*,* d *the* disbelief*,* u *the* uncertainty*, and* a *the atomicity.*

[7] It holds $\alpha_A = r + r_0$ and $\beta_A = s + s_0$ and $(r, r_0, s, s_0) = m_B^{CT}(o_A)$ (analogous for α_B and β_B, for $\alpha_{A \wedge B}$ and $\beta_{A \wedge B}$, and for $\alpha_{A \vee B}$ and $\beta_{A \vee B}$).

[8] We have developed a Java application for the visualization of opinions (also calculating the color-gradient of the background) and for demonstrating the operators. The examples are basically screen shots from this application.

Table 2. Examples for the operators AND and OR

$o_A = (t_A, c_A, f_A)$	$o_B = (t_B, c_B, f_B)$	$o_{A \wedge B}$	$o_A = (t_A, c_A, f_A)$	$o_B = (t_B, c_B, f_B)$	$o_{A \vee B}$
$(0.5,0,0.5)$	$(0.5, 0, 0.5)$	$(0.5, 0, 0.25)$	$(0.5,0,0.5)$	$(0.5, 0, 0.5)$	$(0.5, 0, 0.75)$
$E(o_A) = 0.5$ $f(p\|\alpha_A, \beta_A) = (1, 1)$	$E(o_B) = 0.5$ $f(p\|\alpha_A, \|\beta_A) = (1, 1)$	$E(o_{A \wedge B}) = 0.25$ $f(p\|\alpha_{A \wedge B}, \beta_{A \wedge B})) = (0.5, 1.5)$	$E(o_A) = 0.5$ $f(p\|\alpha_A, \beta_A) = (1, 1)$	$E(o_B) = 0.5$ $f(p\|\alpha_B, \beta_B) = (1, 1)$	$E(o_{A \vee B}) = 0.75$ $f(p\|\alpha_{A \vee B}, \beta_{A \vee B})) = (1.5, 0.5)$
					
$(0,1,0.5)$	$(0.5, 0, 0.5)$	$(0, 1, 0.25)$	$(0,1,0.25)$	$(0.5, 0, 0.75)$	$(0, 0.0769, 0.8125)$
$E(o_A) = 0$	$E(o_B) = 0.5$	$E(o_{A \wedge B}) = 0$	$E(o_A) = 0$	$E(o_B) = 0.75$	$E(o_{A \vee B}) = 0.75$
					
$(1,1,0.5)$	$(0.5, 0, 0.5)$	$(1, 0.333, 0.25)$	$(1,1,0.25)$	$(0.5, 0, 0.75)$	$(1, 1, 0.8125)$
$E(o_A) = 1$	$E(o_B) = 0.5$	$E(o_{A \wedge B}) = 0.5$	$E(o_A) = 1$	$E(o_B) = 0.75$	$E(o_{A \vee B}) = 1$
					

The mapping between CertainTrust and subjective logic opinions is provided in [36]. The mapping of an opinion in CertainTrust to subjective logic is denoted as a function m_{SL}^{CT},which is defined below:

Definition 7.2 (Mapping CertainTrust to SL)
The mapping from an opinion $o = (t, c, f)$ *in CertainTrust to an opinion* $\omega = (b, d, u, a)$ *in subjective logic is denoted as* $(b, d, u, a) = m_{SL}^{CT}(t, c, f)$ *and defined by* $b = t * c$, $d = (1 - t) * c$, $u = 1 - c$, *and* $a = f$.

The inverse mapping can be given as follows:

Definition 7.3 (Mapping SL to CertainTrust)
The mapping from an opinion $\omega = (b, d, u, a)$ *in subjective logic to an opinion* $o = (t, c, f)$ *in CertainTrust is denoted as* $(t, c, f) = m_{CT}^{SL}(b, d, u, a)$ *and defined by* $c = 1 - u$, $a = f$, *and* $t = \frac{b}{b+d}$ *for* $b + d \neq 0$, *else* $t = 0.5$.

This mapping has the following features:

Theorem 7.1 (Equality of Expectation Values)
It holds that $E(b, d, u, a) = b + ua = E(t, c, f)$, *if* $(b, d, u, a) = m_{SL}^{CT}(t, c, f)$ *(for a proof see [36]).*

Theorem 7.2 (Compliance of operators)
Let A *and* B *be independent propositions. In subjective logic,* ω_A *and* ω_B *are two opinions about proposition* A *and* B*, respectively. Using the mapping functions (i.e.* m_{SL}^{CT} *and* m_{CT}^{SL}*), our operators are fully compliant with the operators for the* normalized versions *of* AND*,* OR*,* NOT *provided for subjective logic in [14]. This means that for* $op \in \{AND, OR\}$ *the first two statements and for* NOT *the last two statements are true:*

1. $o_A = m_{CT}^{SL}(\omega_A)$ *and* $o_B = m_{CT}^{SL}(\omega_B) \Rightarrow \omega_{AopB} = m_{SL}^{CT}(o_{AopB})$
2. $\omega_A = m_{SL}^{CT}(o_A)$ *and* $\omega_B = m_{SL}^{CT}(o_B) \Rightarrow o_{AopB} = m_{CT}^{SL}(\omega_{AopB})$
3. $o_A = m_{CT}^{SL}(\omega_A) \Rightarrow \omega_{\neg A} = m_{SL}^{CT}(o_{\neg A})$
4. $\omega_A = m_{SL}^{CT}(o_A) \Rightarrow o_{\neg A} = m_{CT}^{SL}(\omega_{\neg A})$

The proof is given in Appendix E. As the mapping between opinions in CertainTrust and subjective logic is bijective, this basically means that it is possible to switch between the representations as well as the operators.

Although subjective logic provides capabilities for reasoning under uncertainty, our approach has the following advantages:

- It is based on 3 independent parameters (i.e., t, c, and f) whereas in subjective logic b, d, and u are interrelated by $b + d + u = 1$. This independence of the parameters is considered to be a major advantage from a modeling perspective, as it reduces the number of required parameters to a minimum.
- As our model is based on 3 independent parameters, they can be adjusted separately, e.g., one can increase the certainty (c) without changing the average rating (t). In contrast, in subjective logic it is not possible to change only one of those parameters.
- Our model comes with an intuitive graphical representation supporting the user with two orthogonal axes instead of showing the three parameters of belief, disbelief, and uncertainty using three non-orthogonal axes. Furthermore, the visualization the distribution of the expectation value depending on the initial expectation by using a color-gradient makes our visualisation more expressive than the one of subjective logic.

Finally, based on the mapping between CertainTrust and subjective logic, and following the argumentation provided in [14], the operators for AND and OR calculate the same expectation values as when doing the operation on Beta / Dirichlet probability density functions, however, the variance is not exact, but well approximated.

8 Evaluation of the Use Case

In this section, we show how the operators of CertainLogic can be applied to the use case presented in Section 4. The propositional logic term for evaluating the trustworthiness of the system in the use case has been given as:

$$(A_1 \vee A_2) \wedge (B_1 \vee B_2 \vee B_3) \wedge C$$

For the evaluation, we assume that we have good knowledge about the components of subsystem S_1 (consisting of A_1 and A_2) and subsystem S_2 (consisting of B_1, B_2, and B_3) and that the components are highly available. The opinions o_{A_1} and o_{A_2} as well as the resulting opinion $o_{A_1 \vee A_2} = o_{S_1}$ are given in Table 3(a). The opinions o_{B_1}, o_{B_2}, and o_{B_3} as well as the resulting opinion $o_{B_1 \vee B_2 \vee B_3} = o_{S_2}$ are given in Table 3(b). In both cases, the subsystems are highly trustworthy ($E(o_{S_1}) = 0.9963$ and $E(o_{S_2}) = 0.9964$) and the certainty for both systems is also high ($c_{S_1} = 0.9956$ and $c_{S_1} = 0.9894$).

We show the advantage of the new operators presenting different scenarios regarding the trustworthiness of the atomic component C. Depending on whether the component

Table 3. Resulting opinions for S_1 and S_2

(a) S_1:

o_{A_1}	$(0.90, 0.98, 0.5)$
o_{A_2}	$(0.99, 0.95, 0.5)$
$o_{A_1 \vee A_2} = o_{S_1}$	$(0.9974, 0.9956, 0.75)$

(b) S_2:

o_{B_1}	$(0.9, 0.8, 0.5)$
o_{B_2}	$(0.95, 0.8, 0.5)$
o_{B_3}	$(0.9, 0.9, 0.5)$
$o_{B_1 \vee B_2 \vee B_3} = o_{S_2}$	$(0.9978, 0.9894, 0.875)$

is hosted by the owner of the overall system or by a third party, the certainty about the behavior of this component might be higher or lower.

Here we consider two cases:

- *Case 1:* We assume that the trustworthiness of C is given as $o_C = (0.9, 0.9, 0.5)$ [*high certainty*] or as $o_C = (0.9, 0.1, 0.5)$ [*low certainty*]. For this case, the trustworthiness of the overall system S (consisting of S_1, S_2, and C) are given in Table 4(a). In the first row, we see that the *high certainty in* o_C is also reflected in the resulting opinion ($c_S = 0.9229$), whereas the *low certainty in* o_C is reflected in the resulting opinion ($c_S = 0.3315$) in the second row. In this example, we have different expectation values for o_C (depending on the certainty), and thus also different expectation values for o_S.
- *Case 2:* Here, we assume that the trustworthiness of C is given as $o_C = (0.9, 0.9, 0.9)$ [*high certainty*] or as $o_C = (0.9, 0.1, 0.9)$ [*low certainty*]. Here, both opinions lead to the same expectation value. The expectation value for the trustworthiness of the overall system is also the same (due to the compliance with the standard probabilistic approach). However, in our approach the different values for the certainty in the input parameters are still visible in the final result, for the certainty it holds $c_S = 0.9704$ [*high certainty*] and $c_S = 0.7759$ [*low certainty*] (see Table 4(b)).

Table 4. Resulting opinions for S

(a) *Case 1:*

	o_C	$o_{S_1 \wedge S_2 \wedge C} = o_S$
high certainty	$(0.9, 0.9, 0.5)$	$(0.8978, 0.9229, 0.3281)$ $E(o_S) = 0.8538$
low certainty	$(0.9, 0.1, 0.5)$	$(0.9556, 0.3315, 0.3281)$ $E(o_S) = 0.5361$

(b) *Case 2:*

	o_C	$o_{S_1 \wedge S_2 \wedge C} = o_S$
high certainty	$(0.9, 0.9, 0.9)$	$(0.9028, 0.9704, 0, 5906)$ $E(o_S) = 0.8935$
low certainty	$(0.9, 0.1, 0.9)$	$(0.981, 0.7759, 0.5906)$ $E(o_S) = 0.8935$

9 Conclusion

In this paper, we proposed a novel model for the evaluation of propositional logic terms under uncertainty. The operators for AND and OR have been shown to be associative and commutative, which is essential for the evaluation of propositional logic terms. Additionally, the operators have been shown to be compliant with the standard probabilistic evaluation of propositional logic terms and with subjective logic, which finally provides the justification for the mathematical validity of our model. However, the proposed approach is more expressive than the standard probabilistic approach, and although it is as expressive as subjective logic, it provides simpler representation since it is based on independent parameters and provides a more intuitive and more expressive graphical representation.

Furthermore, it has been shown that the parameters for assessing opinions in CertainLogic can be derived using multiple approaches and source: direct assessment by experts (e.g., by certification authorities), derived from Beta probability density functions based on past experiments and prior knowledge (e.g., suitable for statistical data), or using the results of Bayesian reputation systems (e.g., user feedback).

Finally, we have shown the applicability as well as the benefits of our operators in a use case set in the field of evaluating the trustworthiness of a system in a Cloud Computing scenario. It provides a means (1) to derive the trustworthiness of the overall system based on the knowledge about its components, (2) to take into account multiple criteria (modeled by propositions), and (3) to explicitly model the uncertainty associated to the truth of a proposition, which also allows to model that one could have to rely on a not fully trusted source of information. Thus, we consider this approach an appropriate, expressive, and well-founded tool for the evaluation of the trustworthiness of complex systems.

While we have used the Cloud Computing scenario as a descriptive example, the model could also be used for reasoning under uncertainty in other fields such as those involving contextual information. Such information is also subject to uncertainty; for instance, information collected by sensors.

References

1. ENISA: An SME perspective on cloud computing - survey. Technical report, ENISA (2009)
2. Chow, R., Golle, P., Jakobsson, M., Shi, E., Staddon, J., Masuoka, R., Molina, J.: Controlling data in the cloud: outsourcing computation without outsourcing control. In: Proceedings of the 2009 ACM workshop on Cloud computing security. CCSW '09, ACM (2009) 85–90
3. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: Above the clouds: A berkeley view of cloud computing. University of California, Berkeley (Feb 2009)
4. TCG: Trusted computing group (TCG) (2010)
5. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems **43(2)** (2007) 618–644
6. Heimann, D., Mittal, N.: Availability and reliability modeling for computer systems. Advances in Computers **31** (1990) 175–233
7. Jürgenson, A., Willemson, J.: Computing exact outcomes of multi-parameter attack trees. In: Proceedings of the OTM Confederated International Conferences. (2008) 1036 — 1051
8. Schneier, B.: Attack trees: Modeling security threats. Dr. Dobb's journal **24** (1999) 21 – 29
9. Sadeghi, A.R., Stüble, C.: Property-based attestation for computing platforms: caring about properties, not mechanisms. In: Proceedings of the 2004 Workshop on New Security Paradigms. NSPW '04, ACM (2004) 67–77
10. Chen, L., Landfermann, R., Löhr, H., Rohe, M., Sadeghi, A.R., Stüble, C.: A protocol for property-based attestation. In: Proceedings of the first ACM workshop on Scalable trusted computing. STC '06, ACM (2006) 7–16
11. Varadharajan, V.: A note on trust-enhanced security. IEEE Security and Privacy **7** (2009) 57–59
12. Nagarajan, A., Varadharajan, V.: Dynamic trust enhanced security model for trusted computing platform based services. Future Generation Computer Systems (2010)

13. Jøsang, A.: A logic for uncertain probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **9**(3) (2001) 279–212
14. Jøsang, A., McAnally, D.: Multiplication and comultiplication of beliefs. International Journal of Approximate Reasoning **38(1)** (2005) 19–51
15. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: Proceedings of Hawaii International Conference on System Sciences. (2000)
16. Teacy, W.T.L., Patel, J., Jennings, N.R., Luck, M.: TRAVOS: Trust and reputation in the context of inaccurate information sources. Autonomous Agents and Multi-Agent Systems **12**(2) (2006) 183–198
17. Huynh, T.D., Jennings, N.R., Shadbolt, N.R.: Fire: An integrated trust and reputation model for open multi-agent systems. In: Proceedings of the 16th Eureopean Conference on Artificial Intelligence (ECAI), IOS Press (2004) 18–22
18. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The EigenTrust algorithm for reputation management in p2p networks. In: 12th Int. Conference on World Wide Web, (2003) 640–651
19. Buchegger, S., Le Boudec, J.Y.: A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. In: P2PEcon 2004. (2004)
20. Whitby, A., Jøsang, A., Indulska., J.: Filtering out unfair ratings in bayesian reputation systems. The ICFAIN Journal of Management Research **4(2)** (2005) 48 – 64
21. Jøsang, A., Ismail, R.: The beta reputation system. In: Proceedings of the 15th Bled Conference on Electronic Commerce. (2002)
22. Ries, S.: Extending bayesian trust models regarding context-dependence and user friendly representation. In: Proceedings of the 2009 ACM Symposium on Applied Computing, ACM (2009) 1294–1301
23. Gambetta, D.: Can we trust trust? In: Trust: Making and Breaking Cooperative Relations. Basil Blackwell, New York (1990) 213–237
24. Bolstad, W.M.: Introduction to Bayesian Statistics. John Wiley & Sons, Inc (2004)
25. Schryen, G., Volkamer, M., Ries, S., Habib, S.M.: A formal approach towards measuring trust in distributed systems. In: ACM Symp. on Applied Computing. (2011)
26. Ries, S., Heinemann, A.: Analyzing the robustness of CertainTrust. In: 2nd Joint iTrust and PST Conference on Privacy, Trust Management and Security, Springer (2008) 51 – 67
27. Hang, C.W., Wang, Y., Singh, M.P.: Operators for propagating trust and their evaluation in social networks. In: Proceedings of the 8th International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS). (2009)
28. Kerr, R., Cohen, R.: Smart cheaters do prosper: defeating trust and reputation systems. In: AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, (2009) 993–1000
29. Feldman, M., Chuang, J.: Overcoming free-riding behavior in peer-to-peer systems. SIGecom Exch. **5**(4) (2005) 41–50
30. Wang, Y., Singh, M.P.: Formal trust model for multiagent systems. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI). (2007)
31. Zadeh, L.A.: Fuzzy logic and approximate reasoning. Synthese **30** (1975) 407–428
32. Haenni, R.: Towards a unifying theory of logical and probabilistic reasoning. In: ISIPTA'05, 4th Int. Symposium on Imprecise Probabilities and Their Applications. (2005) 193–202
33. Ries, S.: CertainTrust: A trust model for users and agents. In: Proceedings of the 2007 ACM Symposium on Applied Computing, ACM Press (2007) 1599 – 1604
34. Ries, S., Aitenbichler, E.: Limiting sybil attacks on bayesian trust models in open soa environments. In: Proceedings of the The First International Symposium on Cyber-Physical Intelligence (CPI-09). (2009)
35. Jøsang, A., Hird, S., Faccer, E.: Simulating the effect of reputation systems on e-markets. In: Proceedings of the First Int. Conference on Trust Management (iTrust'03). (2003) 179–194
36. Ries, S.: Trust in Ubiquitous Computing. PhD thesis, Technische Universität Darmstadt (2009)

Appendix

Please recall that it holds $0 \leq t_A, t_B, c_A, c_B \leq 1$ and $0 < f_A, f_B < 1$ (see 5.1).

A Proof: Theorem 5.1

Proof. The proof for $o_{A\vee B} = o_{B\vee A}$ can be carried out componentwise by verifying $t_{A\vee B} = t_{B\vee A}$, $c_{A\vee B} = c_{B\vee A}$, and $f_{A\vee B} = f_{B\vee A}$. Using Table 1(OR) for this can easily be verified.

The proof for $o_{A\wedge B} = o_{B\wedge A}$ can be carried out analogously, using Table 1(AND).

B Proof: Theorem 5.2 (*OR*)

Before proving the theorem, we introduce two Lemmas that we need for the proof.

Lemma B.1 *As a Lemma we proof:* $c_{A\vee B} > 0$ *if* $c_A = 0$ *and* $c_B \neq 0$.

Proof.

$$\begin{aligned}
&0 < c_A + c_B - c_A c_B - \frac{c_A(1-c_B)f_B(1-t_A) + (1-c_A)c_B f_A(1-t_B)}{f_A + f_B - f_A f_B} \\
&\text{Using } c_A = 0 \text{ it holds:} \\
&0 < c_B - \frac{c_B f_A(1-t_B)}{f_A + f_B - f_A f_B} \\
&0 < c_B f_A + c_B f_B - c_B f_A f_B - c_B f_A(1-t_B)
\end{aligned} \tag{3}$$

Which is true as it holds $c_B f_A \geq c_B f_A(1-t_B)$ and $c_B f_A > c_B f_A f_B$.

Lemma B.2 *As a Lemma we proof:* $c_{A\vee B} > 0$ *if* $c_A \neq 0$ *and* $c_B = 0$.

Proof. The proof can be done using the commutativity (Theorem 5.1) and Lemma B.1.

Lemma B.3 *As a Lemma we proof:* $c_{A\vee B} > 0$ *if* $c_A \neq 0$ *and* $c_B \neq 0$.

Proof.

$$0 < c_A + c_B - c_A c_B - \frac{c_A(1-c_B)f_B(1-t_A) + (1-c_A)c_B f_A(1-t_B)}{f_A + f_B - f_A f_B}$$

Expand and reorganize, using $f_A + f_B + f_A f_B > 0$ it holds:

$$c_A f_A + c_B f_A + c_A f_B + c_B f_B + c_A c_B f_A f_B > c_A f_A f_B + c_B f_A f_B + c_A c_B f_A + \\ + c_A c_B f_B + c_A(1-c_B)f_B(1-t_A) + (1-c_A)c_B f_A(1-t_B)$$

Simplify:

$$c_A f_A + c_B f_A + c_A f_B + c_B f_B + c_A c_B f_A f_B > c_A f_A f_B + c_B f_A f_B + \\ + c_A f_B(c_B + (1-c_B)(1-t_A)) + c_B f_A(c_A + (1-c_A)(1-t_B))$$

Using $c_A f_A > c_A f_A f_B$ and $c_B f_B > c_B f_A f_B$ it holds:

$$c_B f_A + c_A f_B + c_A c_B f_A f_B > c_A f_B(c_B + (1-c_B)(1-t_A)) + c_B f_A(c_A + (1-c_A)(1-t_B))$$

Using $c_A f_B(c_B + (1-c_B)(1-t_A)) \leq c_A f_B(c_B + (1-c_B))$
and $c_B f_A(c_A + (1-c_A)(1-t_B)) \leq c_B f_A(c_A + (1-c_A))$ it holds:

$$c_A f_B + c_B f_A + c_A c_B f_A f_B > c_A f_B + c_B f_A$$

Simplify:

$$c_A c_B f_A f_B > 0 \tag{4}$$

Which is true as it holds $f_A, f_B, c_A, c_B \neq 0$.

Lemma B.4 *As a Lemma we proof also:* $c_{A\vee B} = 0$ *if* $c_A = 0$ *and* $c_B = 0$.

Proof. This proof can be carried out using $c_A = 0$ and $c_B = 0$ in $c_{A\vee B}$.

Proof. **Proof of Theorem 5.2 (**OR**)**:

The proof will be carried out componentwise by verifying $t_{(A\vee B)\vee C} = t_{A\vee(B\vee C)}$, $c_{(A\vee B)\vee C} = c_{A\vee(B\vee C)}$, and $f_{(A\vee B)\vee C} = f_{A\vee(B\vee C)}$.

Proof for $f_{(A\vee B)\vee C} = f_{A\vee(B\vee C)}$**:**

$$\begin{aligned} f_{(A\vee B)\vee C} &= f_{A\vee B} + f_C - f_{A\vee B} f_C \\ &= (f_A + f_B - f_A f_B) + f_C - (f_A + f_B - f_A f_B) f_C \\ &= f_A + f_B - f_A f_B + f_C - f_A f_C - f_B f_C + f_A f_B f_C \\ &= f_A + (f_B + f_C - f_B f_C) - f_A(f_B + f_C - f_B f_C) \\ &= f_A + f_{B\vee C} - f_A f_{B\vee C} = f_{A\vee(B\vee C)} \end{aligned} \tag{5}$$

Proof for $c_{(A\vee B)\vee C} = c_{A\vee(B\vee C)}$:

$$\begin{aligned} c_{(A\vee B)\vee C} &= c_{A\vee B} + c_C - c_{A\vee B}c_C - \\ &- \frac{c_{A\vee B}(1-c_B)f_B(1-t_{A\vee B}) + (1-c_{A\vee B})c_B f_{A\vee B}(1-t_B)}{f_{A\vee B} + f_B - f_{A\vee B}f_B} \\ &= \text{... [Expand } t_{A\vee B},\ c_{A\vee B}, \text{ and } f_{A\vee B}]\text{...} \end{aligned} \tag{6}$$

$= (c_C((-1+f_A)(-1+f_B)f_C + (-f_A(-1+f_B)+f_B)t_C) + c_B(-(-1+f_A)f_B(1+(-1+c_C)f_C - c_Ct_C) + t_B(-(-1+c_C)f_C + f_A(1+(-1+c_C)f_C - c_Ct_C))) + c_A(f_A(1+(-1+c_B)f_B - c_Bt_B)(1+(-1+c_C)f_C - c_Ct_C) + t_A((-1+c_C)f_C(-1+c_Bt_B) - (-1+c_B)f_B(1+(-1+c_C)f_C - c_Ct_C))))/(f_A(-1+f_B)(-1+f_C) - f_B(-1+f_C) + f_C)$

$$\begin{aligned} &= \text{... [Concentrate } t_{B\vee C},\ c_{B\vee C}, \text{ and } f_{B\vee C}]\text{...} \\ &= c_A + c_{B\vee C} - c_A c_{B\vee C} - \\ &- \frac{c_A(1-c_{B\vee C})f_B(1-t_A) + (1-c_A)c_{B\vee C}f_A(1-t_{B\vee C})}{f_A + f_{B\vee C} - f_A f_{B\vee C}} \\ &= c_{A\vee(B\vee C)} \end{aligned} \tag{7}$$

Proof for $t_{(A\vee B)\vee C} = t_{A\vee(B\vee C)}$:

For proving $t_{(A\vee B)\vee C} = t_{A\vee(B\vee C)}$, we have to consider that there are two cases for calculating $t_{X\vee Y}$:

- (1) the case $c_{X\vee Y} \neq 0$
- and (2) in the case $c_{X\vee Y} = 0$.

For the proof, we use the observation that it holds $c_{X\vee Y} = 0$ if and only if $c_X = c_Y = 0$ (see Lemmas B.3 and B.4), which leads us to 5 cases, which we prove separately:

Case 0. $c_A \neq 0$, $c_B \neq 0$, and $c_C \neq 0$ or exactly one term out of c_A, c_B, and c_C is equivalent to 0: In this case, it holds $c_{(A\vee B)\vee C} \neq 0$ and $c_{A\vee(B\vee C)} \neq 0$ (using Lemmas B.1 and B.2):

$$\begin{aligned} t_{(A\vee B)\vee C} &= \frac{1}{c_{(A\vee B)\vee C}}(c_{A\vee B}t_{A\vee B} + c_Ct_C - c_{A\vee B}c_Ct_{A\vee B}t_C) \\ &= \text{...[Expand } t_{A\vee B},\ c_{A\vee B}, \text{ and } f_{A\vee B}]\text{...} \end{aligned} \tag{8}$$

$= ((f_A(-1+f_B)(-1+f_C) - f_B(-1+f_C) + f_C)(c_Ct_C + c_Bt_B(1-c_Ct_C) + c_At_A(-1+c_Bt_B)(-1+c_Ct_C)))/(c_C((-1+f_A)(-1+f_B)f_C + (-f_A(-1+f_B)+f_B)t_C) + c_B(-(-1+f_A)f_B(1+(-1+c_C)f_C - c_Ct_C) + t_B(-(-1+c_C)f_C + f_A(1+(-1+c_C)f_C - c_Ct_C))) + c_A(f_A(1+(-1+c_B)f_B - c_Bt_B)(1+(-1+c_C)f_C - c_Ct_C) + t_A((-1+c_C)f_C(-1+c_Bt_B) - (-1+c_B)f_B(1+(-1+c_C)f_C - c_Ct_C))))$

$$\begin{aligned} &= \text{... [Concentrate } t_{B\vee C},\ c_{B\vee C}, \text{ and } f_{B\vee C}]\text{...} \\ &= \frac{1}{c_{A\vee(B\vee C)}}(c_At_A + c_{B\vee C}t_{B\vee C} - c_Ac_{B\vee C}t_At_{B\vee C}) \\ &= t_{A\vee(B\vee C)} \end{aligned} \tag{9}$$

Furthermore, there are four cases to consider:

1. $c_A = 0$, $c_B = 0$, and $c_C \neq 0$
2. $c_A = 0$, $c_B \neq 0$, and $c_C = 0$
3. $c_A \neq 0$, $c_B = 0$, and $c_C = 0$
4. $c_A = c_B = c_C = 0$

Case 1.) $c_A = 0$, $c_B = 0$, and $c_C \neq 0$

$$t_{(A\vee B)\vee C} = \frac{1}{c_{(A\vee B)\vee C}}(c_{A\vee B}t_{A\vee B} + c_C t_C - c_{A\vee B}c_C t_{A\vee B}t_C)$$

Using $c_A = c_B = 0 \Rightarrow\ c_{A\vee B} = 0$ (Lemma B.4) it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{(A\vee B)\vee C}}c_C t_C$$

Using $c_{(A\vee B)\vee C} = c_{A\vee(B\vee C)}$ it holds: (10)

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}c_C t_C$$

Using $c_A = c_B = 0$ and $c_{B\vee C}t_{B\vee C} = c_B t_B + c_C t_C - c_B t_B c_C t_C$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}\left(c_A t_A + c_{B\vee C}t_{B\vee C} - c_A c_{B\vee C} t_A t_{B\vee C}\right)$$

$$t_{(A\vee B)\vee C} = t_{A\vee(B\vee C)}$$

Case 2.) $c_A = 0$, $c_B \neq 0$, and $c_C = 0$

$$t_{(A\vee B)\vee C} = \frac{1}{c_{(A\vee B)\vee C}}(c_{A\vee B}t_{A\vee B} + c_C t_C - c_{A\vee B}c_C t_{A\vee B}t_C)$$

Using $c_C = 0$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{(A\vee B)\vee C}}c_{A\vee B}t_{A\vee B}$$

Using $c_{A\vee B}t_{A\vee B} = c_A t_A + c_B t_B + c_A t_A c_B t_B$ and $c_A = 0$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{(A\vee B)\vee C}}c_B t_B$$

Using $c_{(A\vee B)\vee C} = c_{A\vee(B\vee C)}$ it holds: (11)

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}c_B t_B$$

Using $c_{B\vee C}t_{B\vee C} = c_B t_B + c_C t_C + c_B t_B c_C t_C$ and $c_C = 0$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}c_{B\vee C}t_{B\vee C}$$

Using $c_A = 0$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}(c_A t_A + c_{B\vee C}t_{B\vee C} - c_A t_A c_{B\vee C}t_{B\vee C})$$

$$t_{(A\vee B)\vee C} = t_{A\vee(B\vee C)}$$

Case 3.) $c_A \neq 0$, $c_B = 0$, and $c_C = 0$:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{(A\vee B)\vee C}}(c_{A\vee B}t_{A\vee B} + c_C t_C - c_{A\vee B}c_C t_{A\vee B}t_C)$$

Using $c_C = 0$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{(A\vee B)\vee C}}c_{A\vee B}t_{A\vee B}$$

Using $c_{(A\vee B)\vee C} = c_{A\vee(B\vee C)}$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}c_{A\vee B}t_{A\vee B} \tag{12}$$

Using $c_{A\vee B}t_{A\vee B} = c_A t_A + c_B t_B - c_A t_A c_B t_B$ and $c_B = 0$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}c_A t_A$$

Using $c_{B\vee C} = 0$ (Lemma B.4) it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}(c_A t_A + c_{B\vee C}t_{B\vee C} - c_A c_{B\vee C}t_A t_{B\vee C})$$

$$t_{(A\vee B)\vee C} = t_{A\vee(B\vee C)}$$

Case 4.) $c_A = c_B = c_C = 0$

In this case it holds $c_{(A\vee B)\vee C} = c_{A\vee(B\vee C)} = 0$ (applying the Lemma B.4 two times when calculating $c_{(A\vee B)\vee C}$ or $c_{A\vee(B\vee C)}$, respectively, and thus $c_{(A\vee B)\vee C} = t_{A\vee(B\vee C)} = 0.5$

C Proof: Theorem 5.2 (AND)

For proving $t_{(A\wedge B)\wedge C} = t_{A\wedge(B\wedge C)}$, we have to consider that there are two cases for calculating $t_{X\wedge Y}$:

- (1) the case $c_{X\wedge Y} \neq 0$
- and (2) the case $c_{X\wedge Y} = 0$.

For the proof, we use the observation that it holds $c_{X\wedge Y} = 0$ if and only if $c_X = c_Y = 0$ (see Lemma C.1, C.2, C.3 and C.4), which leads us to 5 cases for the proof, which we prove separately.

Before proving the theorem, we prove the four Lemmas that we need for the proof.

Lemma C.1 *As a Lemma we proof:* $c_{A\wedge B} > 0$ *if* $c_A = 0$ *and* $c_B \neq 0$.

Proof.

$$\begin{aligned} &0 < c_B - \frac{c_B(1-f_A)t_B}{1-f_Af_B} \\ &0 < c_B(1-f_Af_B) - c_B(1-f_A)t_B \\ &c_B(1-f_A)t_B < c_B(1-f_Af_B) \end{aligned} \tag{13}$$

Which is true as it holds $1-f_A < 1-f_Af_B$ *and* $c_Bt_B \leq c_B$.

Lemma C.2 *As a Lemma we proof:* $c_{A\wedge B} > 0$ *if* $c_A \neq 0$ *and* $c_B = 0$.

Proof. The proof can be done using the commutativity (Theorem 5.1) and Lemma C.1.

Lemma C.3 *As a Lemma we proof:* $c_{A\wedge B} > 0$ *if* $c_A \neq 0$ *and* $c_B \neq 0$.

Proof.

$$\begin{aligned} &0 < c_A + c_B - c_Ac_B - \frac{(1-c_A)c_B(1-f_A)t_B + c_A(1-c_B)(1-f_B)t_A}{1-f_Af_B} \\ &0 < c_A(1-f_Af_B) + c_B(1-f_Af_B) - c_Ac_B(1-f_Af_B) - (1-c_A)c_B(1-f_A)t_B - \\ &\quad - c_A(1-c_B)(1-f_B)t_A \end{aligned} \tag{14}$$

To proof this we show that it holds A) and B):

A) $c_A(1-f_Af_B) > c_A(1-c_B)(1-f_B)t_A$
which is true as it holds $1-f_Af_B > 1-f_B$ and $c_A \geq c_A(1-c_B)t_A$.

B) $c_B(1-f_Af_B) > c_Ac_B(1-f_Af_B) + (1-c_A)c_B(1-f_A)t_B$ which is true as it holds (using $1-f_Af_B > 1-f_A$):
$c_Ac_B(1-f_Af_B) + (1-c_A)c_B(1-f_A)t_B$
$< c_Ac_B(1-f_Af_B) + (1-c_A)c_B(1-f_Af_B)t_B$
$< c_B(1-f_Af_B)(c_A + (1-c_A)t_B)$
$< c_B(1-f_Af_B)(c_A + (1-c_A))$
$< c_B(1-f_Af_B)$

Lemma C.4 *As a Lemma we proof:* $c_{A\wedge B} = 0$ *if* $c_A = 0$ *and* $c_B = 0$.

Proof. For the proof use $c_A = 0$ and $c_B = 0$ in $c_{A\wedge B}$.

Proof. **Proof Theorem 5.2 (AND)**

Proof for $f_{(A\wedge B)\wedge C} = f_{A\wedge(B\wedge C)}$:

$$f_{(A\wedge B)\wedge C} = f_A f_B f_C = f_{A\wedge(B\wedge C)} \tag{15}$$

Proof for $c_{(A\wedge B)\wedge C} = c_{A\wedge(B\wedge C)}$:

$$\begin{aligned}
c_{(A\wedge B)\wedge C} &= c_{A\wedge B} + c_C - c_{A\wedge B}c_C - \\
&- \frac{(1-c_{A\wedge B})c_C(1-f_{A\wedge B})t_C + c_{A\wedge B}(1-c_C)(1-f_C)t_{A\wedge B}}{1-f_{A\wedge B}f_C} \\
&= \text{... Expand } t_{A\wedge B},\ c_{A\wedge B},\text{ and } f_{A\wedge B} \text{ ...} \\
&= \text{... Concentrate } t_{B\wedge C},\ c_{B\wedge C},\text{ and } f_{B\wedge C} \text{ ...} \\
&= c_A + c_{B\wedge C} - c_A c_{B\wedge C} - \\
&- \frac{(1-c_A)c_{B\wedge C}(1-f_A)t_{B\wedge C} + c_A(1-c_{B\wedge C})(1-f_{B\wedge C})t_A}{1-f_A f_{B\wedge C}} \\
&= c_{A\wedge(B\wedge C)}
\end{aligned} \tag{16}$$

Proof for $t_{(A\wedge B)\wedge C} = t_{A\wedge(B\wedge C)}$:

Case 0.) $c_A \neq 0$, $c_B \neq 0$, and $c_C \neq 0$.
In this case it holds:

$$\begin{aligned}
t_{(A\wedge B)\wedge C} &= \frac{1}{c_{(A\wedge B)\wedge C}}(c_{A\wedge B}c_C t_{A\wedge B}t_C + \\
&\frac{c_{A\wedge B}(1-c_C)(1-f_{A\wedge B})f_C t_{A\wedge B}}{1-f_{A\wedge B}f_C} + \\
&+ \frac{(1-c_{A\wedge B})c_C f_{A\wedge B}(1-f_C)t_C}{1-f_{A\wedge B}f_C}) \\
&= \text{... Expand } t_{A\wedge B},\ c_{A\wedge B},\text{ and } f_{A\wedge B} \text{ ...} \\
&= \text{... Concentrate } t_{B\wedge C},\ c_{B\wedge C},\text{ and } f_{B\wedge C} \text{ ...} \\
&= \frac{1}{c_{A\wedge(B\wedge C)}}(c_A c_{B\wedge C} t_A t_{B\wedge C} + \\
&+ \frac{c_A(1-c_{B\wedge C})(1-f_A)f_{B\wedge C}t_A}{1-f_A f_{B\wedge C}} + \\
&+ \frac{(1-c_A)c_{B\wedge C}f_A(1-f_{B\wedge C})t_{B\wedge C}}{1-f_A f_{B\wedge C}})
\end{aligned} \tag{17}$$

Furthermore, there are four cases to consider:

1. $c_A = 0$, $c_B = 0$, and $c_C \neq 0$
2. $c_A = 0$, $c_B \neq 0$, and $c_C = 0$
3. $c_A \neq 0$, $c_B = 0$, and $c_C = 0$
4. $c_A = c_B = c_C = 0$

Case 1.) $c_A = c_B = 0$

$$t_{(A\wedge B)\wedge C} = \frac{1}{c_{(A\wedge B)\wedge C}}(c_{A\wedge B}c_C t_{A\wedge B}t_C +$$
$$\frac{c_{A\wedge B}(1-c_C)(1-f_{A\wedge B})f_C t_{A\wedge B} + (1-c_{A\wedge B})c_C f_{A\wedge B}(1-f_C)t_C}{1-f_{A\wedge B}f_C})$$

Using $c_A = c_B = 0$ it holds $c_{A\wedge B} = 0$ (Lemma C.4), and thus:

$$t_{(A\wedge B)\wedge C} = \frac{1}{c_{(A\wedge B)\wedge C}}\frac{c_C f_{A\wedge B}(1-f_C)t_C}{1-f_{A\wedge B}f_C}$$

Using $f_{A\wedge B} = f_A f_B$ it holds:

$$t_{(A\wedge B)\wedge C} = \frac{1}{c_{(A\wedge B)\wedge C}}\frac{c_C f_A f_B(1-f_C)t_C}{1-f_A f_B f_C}$$

Using $\frac{1}{c_{(A\wedge B)\wedge C}} = \frac{1}{c_{A\wedge(B\wedge C)}}$ it holds:

$$t_{(A\wedge B)\wedge C} = \frac{1}{c_{A\wedge(B\wedge C)}}\frac{c_C f_A f_B(1-f_C)t_C}{1-f_A f_B f_C} \tag{18}$$

Expanding with $(1-f_B f_C)$ it holds:

$$t_{(A\wedge B)\wedge C} = \frac{1}{c_{A\wedge(B\wedge C)}}\frac{c_C f_A f_B(1-f_C)t_C(1-f_B f_C)}{(1-f_A f_B f_C)(1-f_B f_C)}$$

Using $c_B = 0$ and $c_{B\wedge C}t_{B\wedge C} = \frac{c_C f_B(1-f_C)t_C}{1-f_B f_C}$ it holds:

$$t_{(A\wedge B)\wedge C} = \frac{1}{c_{A\wedge(B\wedge C)}}\frac{c_{B\wedge C} f_A(1-f_{B\wedge C})t_{B\wedge C}}{1-f_A f_{B\wedge C}}$$

Using $c_A = 0$ it holds:

$$t_{(A\wedge B)\wedge C} = \frac{1}{c_{A\wedge(B\wedge C)}}(c_A c_C t_A t_C +$$
$$\frac{c_A(1-c_{B\wedge C})(1-f_A)f_{B\wedge C}t_A + (1-c_A)c_{B\wedge C}f_A(1-f_{B\wedge C})t_C}{1-f_A f_{B\wedge C}})$$

$$t_{(A\wedge B)\wedge C} = t_{A\wedge(B\wedge C)}$$

Case 2.) $c_A = 0$, $c_B \neq 0$, and $c_C = 0$
(analogous to Case 1.)
Case 3.) $c_A \neq 0$, $c_B = 0$, and $c_C = 0$
(analogous to Case 1.)
Case 4.) $c_A = c_B = c_C = 0$
In this case it holds $c_{(A \wedge B) \wedge C} = c_{A \wedge (B \wedge C)} = 0$ (applying Lemma C.4 two times, when calculating $c_{(A \wedge B) \wedge C}$ and $c_{A \wedge (B \wedge C)}$, respectively, and thus $t_{(A \wedge B) \wedge C} = t_{A \wedge (B \wedge C)} = 0.5$.

D Proof: Theorem 5.3

Proof. We can prove each of the equations in the theorem separately. The detail algebraic simplifications are omitted for first two proofs.

1. $E(o_{A \wedge B}) = E(t_{A \wedge B}, c_{A \wedge B}, f_{A \wedge B})$[Using Definition 5.5]
   $= t_{A \wedge B} * c_{A \wedge B} + (1 - c_{A \wedge B}) * f_{A \wedge B}$[Using Definition 5.2]
   $= \ldots$ [Substitution of $t_{A \wedge B}$, $c_{A \wedge B}$, and $f_{A \wedge B}$ using Table 1 (AND) and algebraic simplifications]
   $= (t_A * c_A + (1 - c_A) * f_A)(t_B * c_B + (1 - c_B) * f_B)$
   $= E(o_A)E(o_B)$[Using Definition 5.1 and Definition 5.2]
2. $E(o_{A \vee B}) = E(t_{A \vee B}, c_{A \vee B}, f_{A \vee B})$[Using Definition 5.4]
   $= t_{A \vee B} * c_{A \vee B} + (1 - c_{A \vee B}) * f_{A \vee B}$[Using Definition 5.2]
   $= \ldots$ [Substitution of $t_{A \vee B}$, $c_{A \vee B}$, and $f_{A \vee B}$ using Table 1 (OR) and algebraic simplifications]
   $= (t_A * c_A + (1 - c_A) * f_A) + (t_B * c_B + (1 - c_B) * f_B) - (t_A * c_A + (1 - c_A) * f_A)(t_B * c_B + (1 - c_B) * f_B)$
   $= E(o_A) + E(o_B) - E(o_A)E(o_B)$
   [Using Definition 5.1 and Definition 5.2]
3. $E(o_{\neg A}) = E(t_{\neg A}, c_{\neg A}, f_{\neg A})$[Using Definition 5.6]
   $= (t_{\neg A} * c_{\neg A}) + (1 - c_{\neg A}) * f_{\neg A}$[Using Definition 5.2]
   $= (1 - t_A) * c_A + (1 - c_A) * (1 - f_A)$[Substitution using Table 1 (NOT)]
   $= c_A - t_A * c_A + 1 - c_A - f_A + f_A * c_A$
   $= 1 - (t_A * c_A + (1 - c_A) * f_A)$
   $= 1 - E(o_A)$[Using Definition 5.1 and Definition 5.2]

E Sketch of Proof: Theorem 7.2

Proof. In the following we provide a proof for Theorem 6.2 for the NOT operator and show sketches for AND and OR.

1. $\omega_{\neg A} = (b_{\neg A}, d_{\neg A}, u_{\neg A}, a_{\neg A})$
   $= (d_A, b_A, u_A, 1 - a_A)$[Using Theorem 6 in [13]]
   $= ((1 - t_A)c_A, t_A c_A, 1 - c_A, 1 - f_A)$[Using $o_A = m^{SL}_{CT}(\omega_A)$]
   $= m^{CT}_{SL}(1 - t_A, c_A, 1 - f_A)$[Using Definition 7.2]
   $= m^{CT}_{SL}(t_{\neg A}, c_{\neg A}, f_{\neg A})$[Using Table 1($NOT$)]
   $= m^{CT}_{SL}(o_{\neg A})$
2. $o_{\neg A} = (t_{\neg A}, c_{\neg A}, f_{\neg A})$
   $= (1 - t_A, c_A, 1 - f_A)$[Using Table 1 ($NOT$)]
   $= m^{SL}_{CT}((1 - t_A)c_A, t_A c_A, 1 - c_A, 1 - f_A)$[Using Definition 7.3]
   $= m^{SL}_{CT}(d_A, b_A, u_A, 1 - a_A)$[Using $\omega_A = m^{CT}_{SL} o_A$]
   $= m^{SL}_{CT}(\omega_{\neg A})$[Using Theorem 6 in [13]]

The proof for the operators AND and OR can be carried out analogous to the proof for NOT. However, here we just provide a sketch of the proofs.

1. $\omega_{A\wedge B} = (b_{A\wedge B}, d_{A\wedge B}, u_{A\wedge B}, a_{A\wedge B})$[Using 7.1]
   $= \ldots$ [Substitution of $b_{A\wedge B}$, $d_{A\wedge B}$, $u_{A\wedge B}$,$a_{A\wedge B}$ with b_A, d_A, u_A, a_A, b_B, d_B, u_B, and a_B ...
   ... as defined by the normal multiplication in [14] ...
   ... applying $o_A = m^{SL}_{CT}(\omega_A)$ and $o_B = m^{SL}_{CT}(\omega_B)$ ...
   ... algebraic simplifications and applying Definition 7.2]
   $= m^{CT}_{SL}(t_{A\wedge B}, c_{A\wedge B}, f_{A\wedge B})$
   $= m^{CT}_{SL}(o_{A\wedge B})$
2. $o_{A\wedge B} = (t_{A\wedge B}, c_{A\wedge B}, f_{A\wedge B})$
   $= \ldots$ [Introduce t_A, c_A,... and replace them by b_A, d_A,...]
   $= m^{SL}_{CT}(b_{A\wedge B}, d_{A\wedge B}, u_{A\wedge B}, a_{A\wedge B})$
   $= m^{SL}_{CT}(\omega_{A\wedge B})$
3. $\omega_{A\vee B} = (b_{A\vee B}, d_{A\vee B}, u_{A\vee B}, a_{A\vee B})$[Using 7.1]
   $= \ldots$ [Introduce b_A, d_A,... and replace them by t_A, c_A,...]
   $= m^{CT}_{SL}(t_{A\vee B}, c_{A\vee B}, f_{A\vee B})$
   $= m^{CT}_{SL}(o_{A\vee B})$
4. $o_{A\vee B} = (t_{A\vee B}, c_{A\vee B}, f_{A\vee B})$[Using 5.1]
   $= \ldots$ [Introduce t_A, c_A,... and replace them by b_A, d_A,...]
   $= m^{SL}_{CT}(b_{A\vee B}, d_{A\vee B}, u_{A\vee B}, a_{A\vee B})$
   $= m^{SL}_{CT}(\omega_{A\vee B})$